

FILEID**INSPRSOLD

J 16

```
1 0001 0 MODULE INSPRSOLD (          ! Parse old form of commands
2 0002 0
3 0003 0          IDENT = 'V04-000'; Parse old form of commands
4 0004 0          ADDRESSING_MODE(INTERNAL = GENERAL)
5 0005 1 BEGIN
6 0006 1
7 0007 1
8 0008 1 ****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 * ALL RIGHTS RESERVED.
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 * TRANSFERRED.
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 * CORPORATION.
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 ****
30 0030 1 *
31 0031 1 ++
32 0032 1 * FACILITY: Install
33 0033 1
34 0034 1 * ABSTRACT:
35 0035 1
36 0036 1 * This module controls command parsing and processing.
37 0037 1
38 0038 1 * ENVIRONMENT:
39 0039 1
40 0040 1 * VAX/VMS operating system.
41 0041 1
42 0042 1 * AUTHOR: Bob Grosso, June 1981
43 0043 1
44 0044 1 * Modified by:
45 0045 1
46 0046 1 * V03-007 MSH0061 Michael S. Harvey 5-Jul-1984
47 0047 1 * Add EXEONLY support.
48 0048 1
49 0049 1 * V03-006 MSH0055 Michael S. Harvey 31-May-1984
50 0050 1 * Prevent infinite loops on command input errors.
51 0051 1
52 0052 1 * V03-005 MSH0015 Michael S. Harvey 6-Mar-1984
53 0053 1 * Clarify error message for /REPLACE command.
54 0054 1
55 0055 1 * V03-004 BLS0256 Benn Schreiber 27-Dec-1983
56 0056 1 * Remove conflicting qualifier checking, now done in cld.
57 0057 1 * Allow /DELETE even if the file has been deleted.
```

58	0058	1	
59	0059	1	V03-003 BLS0245 Benn Schreiber 27-Oct-1983
60	0060	1	Correct severity of FAIL message for cases where WARNING
61	0061	1	is appropriate.
62	0062	1	
63	0063	1	V03-002 RPG0002 Bob Grosso 13-Sep-1983
64	0064	1	Re-activate header resident.
65	0065	1	Enable /NOPURGE for old interface.
66	0066	1	
67	0067	1	V03-001 RPG0001 Bob Grosso 29-Aug-1983
68	0068	1	Print help from help library instead of program text.
69	0069	1	--

```
71 0070 1 %SBTTL 'Declarations';
72 0071 1
73 0072 1
74 0073 1 ! Include files
75 0074 1
76 0075 1
77 0076 1 LIBRARY 'SY$LIBRARY:LIB'; ! VAX/VMS system definitions
78 0077 1
79 0078 1 REQUIRE 'SRC$:INSPREFIX.REQ';
80 0220 1 REQUIRE 'LIB$:INSDEF.R32';
81 0279 1
82 0280 1
83 0281 1
84 0282 1 ! Table of contents
85 0283 1
86 0284 1
87 0285 1 FORWARD ROUTINE
88 0286 1 PROCESS_COMMAND, ! execute the command line
89 0287 1 DISPATCH, ! check for conflicting quals and
90 0288 1 ! dispatch to appropriate routine
91 0289 1 HELP; ! print interactive help
92 0290 1
93 0291 1 EXTERNAL
94 0292 1 INSOLCMD; ! Tables describing the old Install commands
95 0293 1
96 0294 1
97 0295 1 ! External routines
98 0296 1
99 0297 1
100 0298 1 EXTERNAL ROUTINE
101 0299 1 ! Module INSMAIN
102 0300 1 INSSOPENFILE, ! handle file open processing
103 0301 1 INSSCLOSEFILE, ! handle file close
104 0302 1 INSSCHECKPRIV, ! enforce privilege check in case INSTALL is installed
105 0303 1
106 0304 1 INSSCREATE, ! perform CREATE command
107 0305 1 INSSDELETE, ! perform DELETE command
108 0306 1 INSSGLOBAL, ! list global sections
109 0307 1 INSSLIST; ! list known file entries
110 0308 1
111 0309 1 EXTERNAL ROUTINE
112 0310 1 CLISDCLPARSE, ! Parse DCL command
113 0311 1 CLISPRESERVE, ! Check if entity present
114 0312 1 CLISGETVALVE, ! Get value from command line
115 0313 1 LBR$OUTPUTHELP, ! print interactive help
116 0314 1 LIB$GETINPUT, ! Get subcommands
117 0315 1 LIB$PUTOUTPUT, ! echo command line, and print help
118 0316 1 PRVSSETPRIV, ! set privilege bits given ASCII string
119 0317 1 STR$COMPARE, ! compare two strings
120 0318 1 STR$CONCAT, ! concatenate strings
121 0319 1 STR$COPYDX, ! Copy strings by descriptor
122 0320 1 STR$POSITION,
123 0321 1 STR$RIGHT,
124 0322 1 STR$UPCASE;
125 0323 1
126 0324 1 EXTERNAL LITERAL
127 0325 1 CLIS_NEGATED, ! Qualifier was explicitly negated
```

Declarations

```

128 0326 1 PRVS_INVNAM,          ! Invalid privilege name
129 0327 1 PRVS_NOTUNQ;        ! Privilege name not unique
130
131 0329 1 EXTERNAL LITERAL
132 0330 1 INSS_EXISTS,          ! Known File Entry already exists
133 0331 1 INSS_FAIL,           ! Unable to <OPERATION> <FILENAME>
134 0332 1 INSS_FAILED,          ! Failed to process request for <FILENAME>
135 0333 1 INSS_HELP,            ! Enter /HELP for INSTALL help
136 0334 1 INSS_INTRNLERR,       ! Internal Error: error detail
137 0335 1 INSS_INVPRVNAM,       ! Invalid privilege name
138 0336 1 INSS_NOKFEFND,        ! Known File not found
139 0337 1 INSS_NOLOAD,          ! Network and sequentially loaded files are not installable
140 0338 1 INSS_PRVNOTUNQ,        ! Privilege name not unique
141 0339 1 INSS_REMOVED;         ! Known entry was removed
142
143 0341 1 EXTERNAL
144 0342 1 INSSL_INTRNLERR,       ! Store internal error descriptor
145 0343 1 INSSG_KFENAM,          ! file name block declaration
146 0344 1 INSSGE_KFECHAN,        ! channel for open known file
147 0345 1 INSSGL_CTLMSK : $BBLOCK, ! Control flags
148 0346 1 INSSGL_REPLACE_MSK: $BBLOCK, ! Control flags saved for REPLACE
149 0347 1 INSSGQ_KFERN$ :        ! buffer for resultant name
150 0348 1 $BBLOCK,
151 0349 1 INSSGL_KFEADR,         ! id of file to be listed
152 0350 1 INSSGQ_KFEPRIVS : $BBLOCK, ! Quadword privilege mask
153 0351 1 INSSGQ_FILDSC :        ! descriptor to get filename returned
154 0352 1 $BBLOCK,
155 0353 1 INSSG_KFEFAB : $BBLOCK, ! from CLI.
156 0354 1 CTLSGQ_PROCPRI : $BBLOCK; ! file access block declaration
157
158 0355 1 CTLSGQ_PROCPRI : $BBLOCK; ! privilege mask of user
159
160 0357 1 OWN
161 0358 1 INPUT_DESC : DYN_DESC_DECL; ! command string input
162
163 0360 1 BIND
164 0361 1 EXIT_DSC = $DESCRIPTOR ('EXIT'),
165 0362 1 FILE_SPEC_DSC = $DESCRIPTOR ('FILE SPEC'),
166 0363 1 GLOBAL_DSC = $DESCRIPTOR ('GLOBAL'),
167 0364 1 LIST_DSC = $DESCRIPTOR ('LIST'),
168 0365 1 PRIVILEGED_DSC = $DESCRIPTOR ('PRIVILÉGED'),
169 0366 1 CREATE_DSC = $DESCRIPTOR ('CREATE'),
170 0367 1 REPLACE_DSC = $DESCRIPTOR ('REPLACE'),
171 0368 1 DELETE_DSC = $DESCRIPTOR ('DELETE'),
172 0369 1 SLASH_REL_DSC = $DESCRIPTOR ('/REL'),
173 0370 1 SLASH_HELP_DSC = $DESCRIPTOR ('/HELP'),
174 0371 1 HEL_DSC = $DESCRIPTOR ('HEL'),
175 0372 1 HELP_DSC = $DESCRIPTOR ('HELP');

```

```
177 0374 1 %SBTTL 'INSSPARSE_OLD_CMD';
178 0375 1
179 0376 1 GLOBAL ROUTINE INSSPARSE_OLD_CMD (PARSE_CMD_DESC, INTERACTIVE) =
180 0377 2 BEGIN
181 0378 2 ***+
182 0379 2
183 0380 2 Command loop. If not interactive, execute command and exit, else
184 0381 2 print "INSTALL>", read in command, parse command and execute command.
185 0382 2 Exit loop for "Z" and /EXIT
186 0383 2
187 0384 2 ---
188 0385 2 MAP
189 0386 2 PARSE_CMD_DESC : REF $BBBLOCK;
190 0387 2 DO
191 0388 2 BEGIN
192 0389 3 LOCAL
193 0390 3 HLP_PTR,
194 0391 3 HLP_LEN,
195 0392 3 STATUS;
196 0393 3
197 0394 3
198 0395 3 IF .INTERACTIVE
199 0396 3 THEN
200 0397 4 BEGIN
201 0398 4 LOCAL
202 0399 4 COPY_DESC : $BBBLOCK [DSC$C_S_BLN],
203 0400 4 DASH_PTR, ! Locate the '-' continuation character
204 0401 4 LAST_WAS_CONT; ! Last command line was continued
205 0402 4
206 0403 4 ! print the prompt and get input line
207 0404 4 STATUS = LIB$GET_INPUT
208 0405 4 ( INPUT_DESC, $DESCRIPTOR('INSTALL> ') );
209 0406 4 IF .STATUS EQL RMSS$EOF ! EXIT on CTRL Z
210 0407 4 THEN EXITLOOP;
211 0408 4 IF NOT .STATUS
212 0409 4 THEN
213 0410 5 BEGIN
214 0411 5 SIGNAL (.STATUS);
215 0412 5 RETURN .STATUS;
216 0413 4 END;
217 0414 4
218 0415 4 DASH_PTR = CH$FIND CH( .INPUT_DESC [DSC$W_LENGTH],
219 0416 4 .INPUT_DESC [DSC$A_POINTER], %('-' );
220 0417 4 IF NOT CH$FAIL (.DASH_PTR)
221 0418 4 THEN
222 0419 5 BEGIN ! Line continued
223 0420 5 LAST_WAS_CONT = TRUE;
224 0421 5 COPY_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
225 0422 5 COPY_DESC [DSC$W_LENGTH] = .DASH_PTR = .INPUT_DESC [DSC$A_POINTER];
226 0423 5 COPY_DESC [DSC$A_POINTER] = .INPUT_DESC [DSC$A_POINTER];
227 0424 5 STR$COPY DX ( .PARSE_CMD_DESC, COPY_DESC ); ! copy dst from src
228 0425 5 WHILE .LAST_WAS_CONT DO
229 0426 5
230 0427 5 ! while last was continued, input next line,
231 0428 5 ! and concatenate with previous.
232 0429 5
233 0430 6 BEGIN
```

```
234 0431 6 STATUS = LIB$GET INPUT( INPUT_DESC, $DESCRIPTOR ('_> ') ); ! get input line
235 0432 6 IF .STATUS EQZ RMSS_EOF
236 0433 6 THEN
237 0434 7 BEGIN
238 0435 7 INTERACTIVE = FALSE;
239 0436 7 EXITLOOP;
240 0437 6 END;
241 0438 6
242 0439 6 COPY_DESC [DSCSA_POINTER] = .INPUT_DESC [DSCSA_POINTER];
243 0440 6
244 0441 6 DASH_PTR = CH$FIND CH( .INPUT_DESC [DSCSW_LENGTH], .INPUT_DESC [DSCSA_POINTER], %c'-');
245 0442 6 IF CH$FAIL (.DASH_PTR)
246 0443 6 THEN
247 0444 7 BEGIN
248 0445 7 LAST_WAS_CONT = FALSE;
249 0446 7 COPY_DESC [DSCSW_LENGTH] = .INPUT_DESC [DSCSW_LENGTH];
250 0447 7 END
251 0448 6 ELSE
252 0449 7 BEGIN
253 0450 7 COPY_DESC [DSCSW_LENGTH] = .DASH_PTR - .INPUT_DESC [DSCSA_POINTER];
254 0451 6 END;
255 0452 6 STR$CONCAT (.PARSE_CMD_DESC, .PARSE_CMD_DESC, COPY_DESC );
256 0453 5 END; ! While reading lines following continuation lines
257 0454 5
258 0455 5 IF NOT .INTERACTIVE THEN EXITLOOP; ! ^Z entered on continuation line
259 0456 5 STR$UPCASE ( .PARSE_CMD_DESC, .PARSE_CMD_DESC );
260 0457 5 END ! If line continued
261 0458 4 ELSE
262 0459 4 STR$UPCASE ( .PARSE_CMD_DESC, INPUT_DESC ); ! Upcase and copy to PARSE_CMD_DESC
263 0460 3 END; ! If interactive
264
265
266
267
268 0463 3 !+++
269 0464 3
270 0465 3 Parse command if there is something on the command line
271 0466 3 ---+
272 0467 3 IF ( .PARSE_CMD_DESC [DSCSW_LENGTH] NEQ 0 ) AND ! If there is something on the line
273 0468 3 NOT CH$FAIL( CH$FIND NOT CH ! and there is something besides blanks
274 0469 3 ( .PARSE_CMD_DESC [DSCSW_LENGTH],
275 0470 3 .PARSE_CMD_DESC [DSCSA_POINTER], %(' ') )
276 0471 3 THEN BEGIN
277 0472 3
278 0473 4 !+++
279 0474 4
280 0475 4 Check for /HELP request
281 0476 4 ---+
282 0477 4
283 0478 4
284 0479 4
285 0480 4
286 0481 4 HLP_PTR = STR$POSITION ( .PARSE_CMD_DESC, SLASH_HELP_DSC); ! Look for "/HELP" on command
287 0482 4 HLP_LEN = 5;
288 0483 4 IF .HLP_PTR EQZ 0 ! If not found, look for "/H
289 0484 4 THEN
290 0485 5 BEGIN
291 0486 5 HLP_PTR = STR$POSITION ( .PARSE_CMD_DESC, SLASH_HEL_DSC);
292 0487 5 HLP_LEN = 4;
```

```
291 0488 4      END;
292 0489 4      IF .HLP_PTR NEQ 0
293 0490 4      THEN
294 0491 4      |
295 0492 4      | Help was requested, treat entire line as help request
296 0493 4      |
297 0494 5      BEGIN
298 0495 5      |
299 0496 5      | replace "/HEL" or "/HELP" with "INSTALL" since lbr$output_help
300 0497 5      | needs to see the topic keyword.
301 0498 5      | Then call the help routine.
302 0499 5      |
303 0500 5      STR$RIGHT ( .PARSE_CMD_DESC, .PARSE_CMD_DESC, %REF (.HLP_PTR+.HLP_LEN) );
304 0501 5      STR$CONCAT (.PARSE_CMD_DESC, $DESCRIPTOR ('INSTALL '), .PARSE_CMD_DESC);
305 0502 5      | Stick "INSTALL on front of help line
306 0503 5      REPORT( HELP ( .PARSE_CMD_DESC ) );
307 0504 5      END
308 0505 5      | Help was requested
309 0506 4      |
310 0507 5      ELSE
311 0508 5      BEGIN
312 0509 5      | Command was something other than /HELP
313 0510 5      |
314 0511 5      | No help requested, proceed to parsing command line
315 0512 5      | Must concatenate command line to "INSTALL" string for
316 0513 5      | cli$dcl_parse to execute.
317 0514 5      |
318 0515 5      |
319 0516 5      STR$CONCAT (.PARSE_CMD_DESC, $DESCRIPTOR ('INSTALL '), .PARSE_CMD_DESC); ! Stick "INSTALL on front
320 0517 5      STATUS = CLI$DCL_PARSESET .PARSE_CMD_DESC, INSOOLDCMD ); ! parse the command line
321 0518 5      |
322 0519 5      IF .STATUS
323 0520 5      THEN
324 0521 6      BEGIN
325 0522 6      IF CLISPRESNT( EXIT_DSC ) ! If /EXIT, then exit program
326 0523 6      THEN EXITLOOP;
327 0524 6      |
328 0525 6      |
329 0526 6      |
330 0527 6      | Set flags for the qualifiers present in command line
331 0528 6      |
332 0529 6      |
333 0530 6      |
334 0531 6      INSSGL_CTLMSK = 0; ! clear flags
335 0532 6      INSSL_INTRNLERR = 0; ! clear internal error descriptor
336 0533 6      |
337 0534 6      | INSTALL control flags
338 0535 6      |
339 0536 6      INSSGL_CTLMSK [INSSV_FILSPC] = CLISPRESNT( FILE_SPEC_DSC );
340 0537 6      |
341 0538 6      INSSGL_CTLMSK [INSSV_CREATE] = CLISPRESNT( CREATE_DSC );
342 0539 6      INSSGL_CTLMSK [INSSV_DELETE] = CLISPRESNT( DELETE_DSC );
343 0540 6      INSSGL_CTLMSK [INSSV_REPLACE] = CLISPRESNT( REPLACE_DSC );
344 0541 6      INSSGL_CTLMSK [INSSV_LIST] = CLISPRESNT( LIST_DSC );
345 0542 6      INSSGL_CTLMSK [INSSV_FULL] = CLISPRESNT( %ASCID 'FULL' );
346 0543 6      INSSGL_CTLMSK [INSSV_GLOBAL] = CLISPRESNT( GLOBAL_DSC );
347 0544 6      INSSGL_CTLMSK [INSSV_HELP] = CLISPRESNT( %ASCID 'HELP' );
```

```

348 0545 6 INSSGL_CTLMSK [INSSV_EXIT] = CLISPRESENT( %ASCID 'EXIT' );
349 0546 6 INSSGL_CTLMSK [INSSV_PROTECT] = CLISPRESENT( %ASCID 'PROTECTED' );
350 0547 6 INSSGL_CTLMSK [INSSV_OPEN] = CLISPRESENT( %ASCID 'OPEN' );
351 0548 6 INSSGL_CTLMSK [INSSV_HDRRES] = CLISPRESENT( %ASCID 'HEADER_RESIDENT' );
352 0549 6 INSSGL_CTLMSK [INSSV_PRIV] = CLISPRESENT( PRIVILEGED_DSC );
353 0550 6 INSSGL_CTLMSK [INSSV_SHARED] = CLISPRESENT( %ASCID 'SHARED' );
354 0551 6 INSSGL_CTLMSK [INSSV_WRITABLE] = CLISPRESENT( %ASCID 'WRITEABLE' );
355 0552 6 INSSGL_CTLMSK [INSSV_NOPURGE] = CLISPRESENT( %ASCID 'NOPURGE' );
356 0553 6 INSSGL_CTLMSK [INSSV_EXEONLY] = CLISPRESENT( %ASCID 'EXECUTE_ONLY' );
357 0554 6
358 0555 6
359 0556 6
360 0557 6
361 0558 6 REPORT (PROCESS_COMMAND( ) );
362 0559 5 END: ! If it parsed
363 0560 5
364 0561 4 END: ! Else command was something other than /HELP
365 0562 3 END: ! If the command line was non-blank
366 0563 3 ! while
367 0564 2 WHILE .INTERACTIVE:
368 0565 2 RETURN TRUE;
369 0566 1 END: ! Routine Parse_old_cmd

```

```

.TITLE INSPRSOLD
.IDENT \V04-000\
.PSECT SPLITS,NOWRT,NOEXE,2

      54 49 58 45 00000 P.AAB: .ASCII \EXIT\
      00000004 00004 P.AAA: .LONG 4
      00000000 00008 P.AAD: .ADDRESS P.AAB
      00015 P.AAC: .ASCII \FILE_SPEC\
      00000009 00018 P.AAF: .BLKB 3
      00000000 0001C P.AAF: .LONG 9
      00026 P.AAE: .ADDRESS P.AAD
      00000006 00028 P.AAE: .ASCII \GLOBAL\
      00000000 0002C P.AAH: .BLKB 2
      54 53 49 4C 00030 P.AAH: .LONG 6
      00000004 00034 P.AAG: .ASCII \LIST\
      00000000 00038 P.AAG: .LONG 4
      00046 P.AAJ: .ADDRESS P.AAH
      00003C P.AAJ: .ASCII \PRIVILEGED\
      000046 P.AAI: .BLKB 2
      0000000A 00048 P.AAI: .LONG 10
      00000000 0004C P.AAI: .ADDRESS P.AAJ
      45 54 41 45 52 43 00050 P.AAL: .ASCII \CREATE\
      00056 P.AAK: .BLKB 2
      00000006 00058 P.AAK: .LONG 6
      00000000 0005C P.AAN: .ADDRESS P.AAL
      45 43 41 4C 50 45 52 00060 P.AAN: .ASCII \REPLACE\
      00067 P.AAM: .BLKB 1
      00000007 00068 P.AAM: .LONG 7
      00000000 0006C P.AAM: .ADDRESS P.AAN
      45 54 45 4C 45 44 00070 P.AAP: .ASCII \DELETE\
      00076 P.AAP: .BLKB 2

```

00000006.	00078	P.AAO:	.LONG 6
00000000.	0007C	P.AAP:	.ADDRESS P.AAP
4C 45 48 2F	00080	P.AAR:	.ASCII \HELP\
00000004.	00084	P.AAQ:	.LONG 4
00000000.	00088	P.AAT:	.ADDRESS P.AAR
50 4C 45 48 2F	0008C	P.AAV:	.ASCII \HELP\
00000005.	00091	P.AAS:	.BLKB 3
00000000.	00094	P.AAV:	.LONG 5
4C 45 48	00098	P.AAV:	.ADDRESS P.AAT
00000003.	0009F	P.AAV:	.ASCII \HELP\
00000000.	000A0	P.AAU:	.BLKB 1
50 4C 45 48	000A4	P.AAV:	.LONG 3
00000004.	000A8	P.AAX:	.ADDRESS P.AAV
00000000.	000AC	P.AAW:	.ASCII \HELP\
20 3E 4C 4C 41 54 53 4E 49	00080	P.AAX:	.LONG 4
00000000.	00084	P.AAZ:	.ADDRESS P.AAX
00000009.	000BD	P.AAZ:	.ASCII \INSTALL> \
00000000.	000C0	P.AAY:	.BLKB 3
20 3E 5F	000C4	P.AAY:	.LONG 9
00000000.	000C8	P.ABB:	.ADDRESS P.AAZ
00000003.	000CB	P.ABB:	.ASCII _> \
00000000.	000CC	P.ABA:	.BLKB 1
20 4C 4C 41 54 53 4E 49	000D0	P.ABA:	.LONG 3
00000008.	000D4	P.ABD:	.ADDRESS P.ABB
00000000.	000DC	P.ABC:	.ASCII \INSTALL \
20 4C 4C 41 54 53 4E 49	000E0	P.ABD:	.LONG 8
00000008.	000E4	P.ABF:	.ADDRESS P.ABD
00000000.	000EC	P.ABE:	.ASCII \INSTALL \
4C 4C 55 46	000F0	P.ABF:	.LONG 8
010E0004.	000F4	P.ABH:	.ADDRESS P.ABF
00000000.	000F8	P.ABG:	.ASCII \FULL\
50 4C 45 48	00100	P.ABH:	.LONG 17694724
010E0004.	00104	P.ABI:	.ADDRESS P.ABH
00000000.	00108	P.ABI:	.ASCII \HELP\
54 49 58 45	0010C	P.ABL:	.LONG 17694724
010E0004.	00110	P.ABL:	.ADDRESS P.ABL
00000000.	00114	P.ABL:	.ASCII \EXIT\
00 00 00 44 45 54 43 45 54 4F 52 50	00118	P.ABN:	.LONG 17694724
010E0009.	00124	P.ABM:	.ADDRESS P.ABN
00000000.	00128	P.ABM:	.ASCII \PROTECTED\<0><0><0>
4E 45 50 4F	0012C	P.ABP:	.LONG 17694729
010E0004.	00130	P.ABO:	.ADDRESS P.ABP
00000000.	00134	P.ABO:	.ASCII \OPEN\
54 4E 45 44 49 53 45 52 5F 52 45 44 41 45 48	00138	P.ABR:	.LONG 17694724
00	00147	P.ABR:	.ADDRESS P.ABR
010E000F.	00148	P.ABR:	.ASCII \HEADER_RESIDENT\<0>
00000000.	0014C	P.ABR:	.LONG 17694735
00 00 44 45 52 41 48 53	00150	P.ABT:	.ADDRESS P.ABR
010E0006.	00158	P.ABT:	.ASCII \SHARED\<0><0>
00000000.	0015C	P.ABT:	.LONG 17694726
00 00 00 45 4C 42 41 45 54 49 52 57	00160	P.ABV:	.ADDRESS P.ABT
010E0009.	0016C	P.ABV:	.ASCII \WRITEABLE\<0><0><0>
00000000.	00170	P.ABU:	.LONG 17694729
00 45 47 52 55 50 4F 4E	00174	P.ABX:	.ADDRESS P.ABU
010E0007	0017C	P.ABX:	.ASCII \NOPURGE\<0>

```

59 4C 4E 4F 5F 45 54 55 43 45 00000000' 00180 .ADDRESS P.ABX
58 45 00184 P.ABZ: .ASCII \EXECUTE_ONLY\
010E000C 00190 P.ABY: .LNG 17694732
00000000' 00194 .ADDRESS P.ABZ

.PSECT $OWNS,NOEXE,2

00# 00000 INPUT_DESC:
02 0E 00002 .BYTE 0[2]
00004 .BYTE 14, 2
00004 .BLKB 4

EXIT_DSC= P.AAA
FILE_SPEC_DSC= P.AAC
GLOBAL_DSC= P.AAE
LIST_DSC= P.AAG
PRIVILEGED_DSC= P.AAI
CREATE_DSC= P.AAK
REPLACE_DSC= P.AAM
DELETE_DSC= P.AAO
SLASH_REL_DSC= P.AAQ
SLASH_HELP_DSC= P.AAS
HELP_DSC= P.AAU
HELP_DSC= P.AAW

.EXTRN IN$OLDCMD, IN$OPEN FILE
.EXTRN IN$CLOSE FILE, IN$CHECK_PRIV
.EXTRN IN$CREATE, IN$DELETE
.EXTRN IN$GLOBAL, IN$LIST
.EXTRN CLISDCL_PARSE, CLISPRES
.EXTRN CLISGET_VALUE, LBR$OUTPUT HELP
.EXTRN LIB$GET_INPUT, LIB$PUT_OUTPUT
.EXTRN PRV$SETPRIV, STR$COMPARE
.EXTRN STR$CONCAT, STR$COPY DX
.EXTRN STR$POSITION, STR$RIGHT
.EXTRN STR$UPCASE, CLIS_NEGATED
.EXTRN PRV$_INVNAME, PRV$_NOTUNQ
.EXTRN IN$EXISTS, IN$FAIL
.EXTRN IN$FAILED, IN$HELP
.EXTRN IN$INTRNLERR, IN$INVPRVNAM
.EXTRN IN$NOKEFND, IN$NOLOAD
.EXTRN IN$PRVNOTUNQ, IN$REMOVED
.EXTRN IN$INTRNLERR
.EXTRN IN$G_KFENAM, IN$GL_KFECHAN
.EXTRN IN$G_CTLMSK, IN$G_REPLACE_MSK
.EXTRN IN$G_Q_KFERNS, IN$GL_KFEADR
.EXTRN IN$G_Q_KFEPRIVS
.EXTRN IN$G_Q_FILDSC, IN$G_KFEFAB
.EXTRN CTL$G_Q_PROCPRIV

.PSECT $CODE$,NOWRT,2

OFFC 00000
.ENTRY IN$PARSE_OLD_CMD, Save R2,R3,R4,R5,R6,R7,- ; 0376
R8,R9,R10,R11
5B 0000' CF 9E 00002 MOVAB INPUT_DESC, R11
5A 0000' CF 9E 00007 MOVAB P.AAY, R10
59 00000000G 00 9E 0000C MOVAB IN$GL_CTLMSK, R9
58 00000000G 00 9E 00013 MOVAB CLISPRES, R8

```

			5E	0C	C2	0001A	SUBL2	#12 SP					
			57	AC	DD	0001D	MOVL	INTÉRACTIVE, R7			0395		
			52	AC	DD	00021	MOVL	PARSE_CMD_DESC, R2			0468		
			03	57	E8	00025	BLBS	R7 1\$			0395		
				00C5	31	00028	BRW	14\$					
					5A	DD	0002B	1\$:	PUSHL	R10			
					5B	DD	0002D		PUSHL	R11			
			00000000G	00	02	FB	0002F	CALLS	#2, LIB\$GET_INPUT				
			0001827A	8F	55	DD	00036	MOVL	R0, STATUS				
					55	D1	00039	CMPL	STATUS, #98938				
					03	12	00040	BNEQ	2\$				
					022F	31	00042	BRW	23\$				
						55	E8	00045	2\$:	BLBS	STATUS, 3\$		
						55	DD	00048		PUSHL	STATUS		
			00000000G	00	01	FB	0004A	CALLS	#1, LIB\$SIGNAL				
			0001827A	8F	50	DD	00051	MOVL	STATUS, R0				
						55	04	00054	RET				
					63	53	AB	00055	3\$:	MOVL	INPUT_DESC+4, R3		
						68	2D	3A	00059	LOCC	#45, INPUT_DESC, (R3)		
							02	12	0005D	BNEQ	4\$		
							51	D4	0005F	CLRL	R1		
						54	51	D0	00061	4\$:	MOVL	R1, DASH_PTR	
						52	AC	DD	00064	MOVL	PARSE_CMD_DESC, R2		
							54	D5	00068	TSTL	DASH_PTR		
							79	13	0006A	BEQL	12\$		
						04	01	D0	0006C	MOVL	#1, LAST_WAS_CONT		
							07	AE	01	90	0006F	MOVB	#1, COPY_DESC+3
							54	53	A3	00073	SUBW3	R3, DASH_PTR, COPY_DESC	
							08	AE	53	DD	00078	MOVL	R3, COPY_DESC+4
										PUSHAB	COPY_DESC		
										PUSHL	R2		
						00000000G	00	02	FB	00081	CALLS	#2, STR\$COPY DX	
						4F	56	E9	00088	5\$:	BLBC	LAST_WAS_CONT, 10\$	
								AA	9F	0008B	PUSHAB	P.ABA	
								58	DD	0008E	PUSHL	R11	
						00000000G	00	02	FB	00090	CALLS	#2, LIB\$GET_INPUT	
						55	50	DD	00097	MOVL	R0, STATUS		
						0001827A	8F	55	D1	0009A	CMPL	STATUS, #98938	
								05	12	000A1	BNEQ	6\$	
								08	AC	D4	000A3	CLRL	INTERACTIVE
								32	11	000A6	BRB	10\$	
								53	DD	000AC	MOVL	R3, COPY_DESC+4	
								68	3A	000B0	LOCC	#45, INPUT_DESC, (R3)	
								02	12	000B4	BNEQ	7\$	
								51	D4	000B6	CLRL	R1	
								54	D0	000B8	7\$:	MOVL	R1, DASH_PTR
								08	12	000BB	BNEQ	8\$	
								56	D4	000BD	CLRL	LAST_WAS_CONT	
								68	B0	000BF	MOVW	INPUT_DESC, COPY_DESC	
								05	11	000C3	BRB	9\$	
								53	A3	000C5	8\$:	SUBW3	R3, DASH_PTR, COPY_DESC
								54	9F	000CA	9\$:	PUSHAB	COPY_DESC
								52	DD	000CD	PUSHL	R2	
								52	DD	000CF	PUSHL	R2	
								03	FB	000D1	CALLS	#3, STR\$CONCAT	
								AE	11	000D8	BRB	5\$	

INSPRSOLD
V04-000

INSSPARSE_OLD_CMD

J 1
16-Sep-1984 01:47:51 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:35:40 [INSTAL.SRC]INSPRSOLD.B32;1

Page 12
(3)

IN
VO

69	01	68	01	FB 0019A	CALLS #1, CLISPRESENT	
		04	50	FO 0019D	INSV R0, #4, #1, INSSGL_CTLMSK	0539
69	01	68	AA	9F 001A2	PUSHAB DELETE_DSC	
		06	01	FB 001A5	CALLS #1, CLISPRESENT	0540
69	01	68	50	FO 001A8	INSV R0, #6, #1, INSSGL_CTLMSK	
		05	AA	9F 001AD	PUSHAB REPLACE_DS	0541
69	01	68	01	FB 001B0	CALLS #1, CLISPRESENT	
		01	50	FO 001B3	INSV R0, #5, #1, INSSGL_CTLMSK	0542
01 A9	01	68	CA	9F 001B8	PUSHAB LIST_DSC	
		01	01	FB 001BC	CALLS #1, CLISPRESENT	0543
01 A9	01	68	50	FO 001BF	INSV R0, #1, #1, INSSGL_CTLMSK+1	
		02	AA	9F 001C5	PUSHAB P.ABG	0544
01 A9	01	68	01	FB 001C8	CALLS #1, CLISPRESENT	
		01	50	FO 001CB	INSV R0, #2, #1, INSSGL_CTLMSK+1	0545
01 A9	01	68	CA	9F 001D1	PUSHAB GLOBAL_DSC	
		04	01	FB 001D5	CALLS #1, CLISPRESENT	0546
01 A9	01	68	50	FO 001D8	INSV R0, #4, #1, INSSGL_CTLMSK+1	
		04	AA	9F 001DE	PUSHAB P.ABI	0547
69	01	68	01	FB 001E1	CALLS #1, CLISPRESENT	
		02	50	FO 001E4	INSV R0, #2, #1, INSSGL_CTLMSK	0548
69	01	68	AA	9F 001E9	PUSHAB P.ABK	
		03	01	FB 001EC	CALLS #1, CLISPRESENT	0549
69	01	68	50	FO 001EF	INSV R0, #3, #1, INSSGL_CTLMSK	
		00	AA	9F 001F4	PUSHAB P.ABM	0550
02 A9	01	68	01	FB 001F7	CALLS #1, CLISPRESENT	
		00	50	FO 00200	INSV R0, #0, #1, INSSGL_CTLMSK+2	0551
01 A9	01	68	AA	9F 00203	PUSHAB P.ABO	
		05	01	FB 00206	CALLS #1, CLISPRESENT	0552
01 A9	01	68	50	FO 0020C	INSV R0, #5, #1, INSSGL_CTLMSK+1	
		06	CA	9F 00210	PUSHAB P.ABQ	0553
01 A9	01	68	01	FB 00213	CALLS #1, CLISPRESENT	
		07	50	FO 00219	INSV R0, #6, #1, INSSGL_CTLMSK+1	0554
01 A9	01	68	AA	9F 0021C	PUSHAB PRIVILEGED_DSC	
		07	01	FB 00225	CALLS #1, CLISPRESENT	0555
01 A9	01	68	50	FO 00229	INSV R0, #7, #1, INSSGL_CTLMSK+1	
		01	CA	9F 00232	PUSHAB P.ABS	0556
02 A9	01	68	01	FB 00236	CALLS #1, CLISPRESENT	
		02	50	FO 00239	INSV R0, #2, #1, INSSGL_CTLMSK+2	0557
02 A9	01	68	CA	9F 0023F	PUSHAB P.ABW	
		03	01	FB 00243	CALLS #1, CLISPRESENT	0558
02 A9	01	68	50	FO 00246	INSV R0, #3, #1, INSSGL_CTLMSK+2	
		03	CA	9F 0024C	PUSHAB P.ABY	0559
02 A9	01	68	01	FB 00250	CALLS #1, CLISPRESENT	
	0000V	05	50	FO 00253	INSV R0, #5, #1, INSSGL_CTLMSK+2	0560
		CF	00	FB 00259	CALLS #C. PROCESS_COMMAND	
		09	50	E8 0025E	21\$: BLBS STATUS, 22\$	0561
	00000000G	00	50	DD 00261	PUSHL STATUS	
		57	01	FB 00263	CALLS #1, LIB\$SIGNAL	0562
		03	AC	DD 0026A	22\$: MOVL INTERACTIVE, R7	
		08	57	E9 0026E	BLBC R7, 23\$	0563
		FDB7	31	00271	BRW 1\$	
		50	01	DO 00274	23\$: MOVL #1, R0	0564
			04	00277	RET	0565
						0566

INSPRSOLD
V04-000

INSSPARSE_OLD_CMD

; Routine Size: 632 bytes, Routine Base: \$CODE\$ + 0000

; 370 0567 1

L 1
16-Sep-1984 01:47:51
14-Sep-1984 12:35:40 VAX-11 Bliss-32 V4.0-742
[INSTAL.SRC]INSPRSOLD.B32;1

Page 14
(3)

IN
VO

```
process_command
: 372 0568 1 %SBTTL 'process_command';
: 373 0569 1
: 374 0570 1 ROUTINE PROCESS_COMMAND =
: 375 0571 2 BEGIN
: 376 0572 2 ***+
: 377 0573 2
: 378 0574 2 FUNCTIONAL DESCRIPTION:
: 379 0575 2
: 380 0576 2 Set up the control flags and call the appropriate routines to execute
: 381 0577 2 the commands.
: 382 0578 2
: 383 0579 2 IMPLICIT INPUT:
: 384 0580 2
: 385 0581 2 CLI$DCL_PARSE has been called with the input command line.
: 386 0582 2
: 387 0583 2 OUTPUT.
: 388 0584 2
: 389 0585 2
: 390 0586 2
: 391 0587 2 IMPLICIT OUTPUT:
: 392 0588 2
: 393 0589 2 Set up global control flags, set up FAB for open file, store
: 394 0590 2 any privilege mask specified and record the open channel number.
: 395 0591 2
: 396 0592 2 ROUTINE VALUE:
: 397 0593 2
: 398 0594 2 success
: 399 0595 2 error:
: 400 0596 2 RMS from file open
: 401 0597 2 PRV$SETPRIV: if there are invalid privileges specified
: 402 0598 2
: 403 0599 2 ---
: 404 0600 2 OWN
: 405 0601 2 PRIV_DSC : $BBLOCK [DSC$C_S_BLN]
: 406 0602 2 PRESET([DSC$B_CLASS] = DSC$K_CLASS_D);
: 407 0603 2 LOCAL
: 408 0604 2 CMK_ARGLST : VECTOR [1], ! arguement list for change mode dispatcher
: 409 0605 2 STATUS;
: 410 0606 2
: 411 0607 2
: 412 0608 2 Since INSTALL may be installed with CMKRNL privilege,
: 413 0609 2 check that user has CMKRNL privilege before allowing
: 414 0610 2 any operations other than /EXIT and /HELP operations.
: 415 0611 2
: 416 0612 2 CMK_ARGLST [0] = 0;
: 417 0613 2 STATUS = $CMKRNL (ROUTIN = INSS$CHECK_PRIV, ARGLST = CMK_ARGLST);
: 418 0614 2 IF NOT .STATUS
: 419 0615 2 THEN
: 420 0616 3 BEGIN
: 421 0617 3 SIGNAL (SSS_NOCMKRNL);
: 422 0618 3 RETURN TRUE;
: 423 0619 2 END;
: 424 0620 2
: 425 0621 2 IF .INSS$GL_CTLMSK [INSS$V_PRIV]
: 426 0622 2 THEN
: 427 0623 3 BEGIN
: 428 0624 3 LOCAL
```

```
429 0625 3      ALL_PRIVS; ! boolean
430 0626 3
431 0627 3      ALL_PRIVS = TRUE;
432 0628 3      CH$FILL (0, 8, INSS$GQ_KFEPRIVS);           ! turn on all privs by default
433 0629 3      WHILE CLIS$GET_VALUE ("PRIVILEGED_DSC", PRIV_DSC) DO
434 0630 4      BEGIN
435 0631 4      ALL_PRIVS = FALSE;
436 0632 4      SELECT PRVS$SETPRIV (PRIV_DSC, INSS$GQ_KFEPRIVS) OF
437 0633 4      SET
438 0634 4
439 0635 4      [PRVS_INVNAM]:
440 0636 5      BEGIN
441 0637 5      SIGNAL (INSS_INVPRVNAM, 1, PRIV_DSC);
442 0638 5      RETURN TRUE;
443 0639 4      END;
444 0640 4
445 0641 4      [PRVS_NOTUNQ]:
446 0642 5      BEGIN
447 0643 5      SIGNAL (INSS_PRVNNOTUNQ, 1, PRIV_DSC);
448 0644 5      RETURN TRUE;
449 0645 4      END;
450 0646 4      TES;
451 0647 3      END;           ! end while there are more privs
452 0648 3      IF .ALL_PRIVS
453 0649 3      THEN CH$FILL (-1, 8, INSS$GQ_KFEPRIVS);           ! No privs specified with /PRIV so
454 0650 2      END;           ! use default of all privs set
455 0651 2
456 0652 2      !++
457 0653 2      | Act on the command entered
458 0654 2
459 0655 2      |---|
460 0656 2
461 0657 2
462 0658 3      IF (NOT .INSS$GL_CTLMSK [INSS$V_FILSPC] )
463 0659 2      THEN
464 0660 2      |
465 0661 2      | A file spec wasn't entered, therefore the only legal qualifiers
466 0662 2      | are /LIST and /GLOBAL and /FULL
467 0663 2
468 0664 3      BEGIN
469 0665 3      LOCAL
470 0666 3      STATUS;
471 0667 3
472 0668 4      IF (.INSS$GL_CTLMSK [INSS$V_LIST] OR
473 0669 4      .INSS$GL_CTLMSK [INSS$V_FULL])
474 0670 3      THEN
475 0671 4      BEGIN
476 0672 4      STATUS = INSS$LIST (0);
477 0673 4      IF NOT .STATUS
478 0674 4      THEN
479 0675 4      SIGNAL (((INSS_FAILED AND NOT STSS$M_SEVERITY) OR STSS$K_WARNING),
480 0676 4                  1, LIST_DSC, .STATUS);
481 0677 4      RETURN TRUE;
482 0678 3      END;
483 0679 3
484 0680 3      IF .INSS$GL_CTLMSK [INSS$V_GLOBAL]
485 0681 3      THEN
```

```

486 0682 4 BEGIN
487 0683 4 STATUS = INSS$GLOBAL ();
488 0684 4 IF NOT .STATUS
489 0685 4 THEN
490 0686 4 SIGNAL (((INSS$_FAILED AND NOT STSS$M_SEVERITY) OR STSS$K_WARNING),
491 0687 4 1, GLOBAL_DSC, :STATUS);
492 0688 4 RETURN TRUE;
493 0689 3 END;
494 0690 3
495 0691 3
496 0692 3
497 0693 3
498 0694 3 A file name was specified
499 0695 3
500 0696 3
501 0697 2 ELSE
502 0698 3 BEGIN
503 0699 3 LOCAL
504 0700 3 OPEN STATUS,
505 0701 3 STATUS;
506 0702 3
507 0703 3
508 0704 3 Get the file spec from the CLI
509 0705 3
510 0706 3 REPORT( CLISGET_VALUE ( FILE_SPEC_DSC, INSS$GQ_FILDSC) );
511 0707 3 OPEN STATUS = INSS$OPEN_FILE ?>;
512 0708 3 IF NOT .OPEN_STATUS
513 0709 3 THEN
514 0710 3 INSS$CLOSE_FILE ();
515 0711 3
516 0712 3 !***+
517 0713 3
518 0714 3 Process file operations
519 0715 3
520 0716 3 !---
521 0717 3 DISPATCH ();
522 0718 3
523 0719 3 INSS$CLOSE_FILE ();
524 0720 3
525 0721 2 END;
526 0722 2 RETURN TRUE;
527 0723 1 END; ! routine process_command

```

```
.PSECT $0WN$,NOEXE,2
```

```
00# 00008 PRIV_DSC:
02 0000B .BYTE 0[3]
0000C .BYTE 2
0000C .BLKB 4
```

```
.EXTRN SYSS$CMKRNL
```

```
.PSECT $CODE$,NOWRT,2
```

```
0FFC 00000 PROCESS_COMMAND:
```


45	52 E8 00005	BLBS	STATUS, 15\$: 0684
0000'	52 DD 00008	PUSHL	STATUS	: 0687
	CF 9F 0000A	PUSHAB	GLOBAL_DSC	: 0686
	01 DD 0000E	PUSHL	#1	
68 00000000*	8F DD 000E0	PUSHL	#<INSS FAILED&-8>	\$D
	04 FB 000E6	CALLS	#4, LIB\$SIGNAL	
	32 11 000E9	BRB	15\$	
00000000G	00 9F 000EB	PUSHAB	INSS\$GQ FIELDSC	0588
00000000G	00 9F 000F1	PUSHAB	FILE SPEC DSC	0706
0000'	02 FB 000F5	CALLS	#2, CLI\$GET_VALUE	:
6B 05	50 E8 000F8	BLBS	STATUS, 13\$	\$D
	50 DD 000FB	PUSHL	STATUS	
00000000G 68	01 FB 000FD	CALLS	#1, LIB\$SIGNAL	
00000000G 00	00 FB 00100	CALLS	#0, INSS\$OPEN_FILE	0707
00000000G 07	00 E8 00107	BLBS	OPEN STATUS, 14\$	0708
00000000G 00	00 FB 0010A	CALLS	#0, INSS\$CLOSE_FILE	0710
0000V CF	00 FB 00111	CALLS	#0, DISPATCH	0717
00000000G 00	00 FB 00116	CALLS	#0, INSS\$CLOSE_FILE	0719
50	01 DD 0011D	MOVL	#1, R0	0722
	04 00120	RET		0723

: Routine Size: 289 bytes. Routine Base: \$CODE\$ + 0278

: 528 0724 1

```

: 530 0725 1 %SBTTL 'dispatch';
: 531 0726 1 ROUTINE DISPATCH =
: 532 0727 1 !++
: 533 0728 1
: 534 0729 1
: 535 0730 1 ---  

: 536 0731 2 BEGIN
: 537 0732 2 LOCAL
: 538 0733 2 STATUS;
: 539 0734 2
: 540 0735 2 IF NOT .INSSG_KFEFAB[FAB$L STS]
: 541 0736 2 AND NOT .INSSGL_CTLMSK[INSSV_DELETE]
: 542 0737 2 THEN RETURN TRUE;
: 543 0738 2 IF .INSSGL_CTLMSK [INSSV_DELETE]
: 544 0739 2 THEN
: 545 0740 2
: 546 0741 2 if /delete requested then /replace/create are invalid
: 547 0742 2 if /list requested then all the remaining known files will
: 548 0743 2 be listed after the delete
: 549 0744 2
: 550 0745 3 BEGIN
: 551 0746 3 STATUS = INSSDELETE (); ! delete it
: 552 0747 3 IF NOT .STATUS
: 553 0748 3 THEN
: 554 0749 4 BEGIN
: 555 0750 4 SIGNAL (((INSS_FAIL AND NOT STSSM_SEVERITY) OR STSSK_WARNING),
: 556 0751 4 2,_DELETE_DSC, INSSGQ_KFERNS, .STATUS);
: 557 0752 4
: 558 0753 4 IF .STATUS EQ INSS_INTRNLERR
: 559 0754 4 THEN
: 560 0755 4 SIGNAL (INSS_INTRNLERR, 1, .INSSL_INTRNLERR);
: 561 0756 4 END
: 562 0757 3 ELSE IF .STATUS EQ INSS_REMOVED
: 563 0758 3 THEN SIGNAL(.STATUS,T,INSSGQ_KFERNS);
: 564 0759 4 IF (.INSSGL_CTLMSK [INSSV_LIST] OR .INSSGL_CTLMSK [INSSV_FULL])
: 565 0760 3 THEN REPORT(INSSLIST (0));
: 566 0761 3 END
: 567 0762 3
: 568 0763 3
: 569 0764 3
: 570 0765 2 ELSE
: 571 0766 2 IF .INSSGL_CTLMSK [INSSV_REPLACE]
: 572 0767 2 THEN
: 573 0768 2
: 574 0769 2 Replace has been requested. There can be no other qualifiers.
: 575 0770 2 The entry must be deleted and recreated.
: 576 0771 2 If /list requested with /replace then do the replace and
: 577 0772 2 list the new entry.
: 578 0773 2
: 579 0774 3 BEGIN
: 580 0775 3
: 581 0776 3 Delete and recreate with new attributes
: 582 0777 3
: 583 0778 3 INSSGL_CTLMSK [INSSV_DELETE] = TRUE; ! set delete flag
: 584 0779 3 INSSGL_REPLACE MSK = 0;
: 585 0780 3 STATUS = INSSDELETE (); ! recreate it
: 586 0781 3 IF NOT .STATUS

```

587 0782 3
588 0783 4
589 0784 6
590 0785 4
591 0786 4
592 0787 4
593 0788 4
594 0789 4
595 0790 4
596 0791 3
597 0792 3
598 0793 3
599 0794 3
600 0795 3
601 0796 3
602 0797 3
603 0798 3
604 0799 3
605 0800 3
606 0801 3
607 0802 3
608 0803 3
609 0804 3
610 0805 3
611 0806 3
612 0807 3
613 0808 3
614 0809 4
615 0810 4
616 0811 4
617 0812 4
618 0813 4
619 0814 4
620 0815 3
621 0816 3
622 0817 4
623 0818 3
624 0819 3
625 0820 3
626 0821 3
627 0822 3
628 0823 3
629 0824 3
630 0825 2
631 0826 2
632 0827 2
633 0828 2
634 0829 2
635 0830 2
636 0831 2
637 0832 2
638 0833 3
639 0834 4
640 0835 3
641 0836 3
642 0837 3
643 0838 4

THEN
BEGIN
SIGNAL (((INSS_FAIL AND NOT STSSM_SEVERITY)
OR STSSK WARNING),
2, DELETE_DSC, INSSGQ_KFERNS, .STATUS);

IF .STATUS EQ INSS_INTRNLERR
THEN
SIGNAL (INSS_INTRNLERR, 1, .INSSL_INTRNLERR);

END;
INSSGL_KFEADR = 0; ! Entry has gone away
INSSCLOSE_FILE();

INSSOPEN FILE();
INSSGL_REPLACE_MSK [INSSV_LIST] = .INSSGL_CTLMSK [INSSV_LIST];
INSSGL_REPLACE_MSK [INSSV_FULL] = .INSSGL_CTLMSK [INSSV_FULL];
INSSGL_CTLMSK = .INSSGL_REPLACE_MSK; ! use replace mask
INSSGL_CTLMSK [INSSV_CREATE] = TRUE; ! set create flag
INSSGL_CTLMSK [INSSV_DELETE] = FALSE; ! clear delete flag
IF .INSSGL_KFEADR EQ 0
THEN
STATUS = INSSCREATE(); ! recreate it
ELSE
STATUS = INSS_EXISTS;

IF NOT .STATUS
THEN
BEGIN
SIGNAL (INSS_FAIL, 2, REPLACE_DSC, INSSGQ_KFERNS, .STATUS);

IF .STATUS EQ INSS_INTRNLERR
THEN
SIGNAL (INSS_INTRNLERR, 1, .INSSL_INTRNLERR);

END;

IF (.INSSGL_CTLMSK [INSSV_LIST] OR .INSSGL_CTLMSK [INSSV_FULL])
THEN REPORT (INSSLIST (.INSSGL_KFEADR));
IF .INSSGL_CTLMSK [INSSV_GLOBAL]
THEN REPORT (INSSGLOBAL (.INSSGL_KFEADR));

END ! /REPLACE

ELSE
filespec given but not for replace or delete.
A /LIST without /CREATE requests list of given file's KFE.
A /LIST with /CREATE requests file be added to KFE and
that entry be listed.
A filename without /LIST or /CREATE is an implied /CREATE.

BEGIN
IF (.INSSGL_CTLMSK [INSSV_LIST] OR .INSSGL_CTLMSK [INSSV_FULL] OR
.INSSGL_CTLMSK [INSSV_GLOBAL]) AND
NOT .INSSGL_CTLMSK [INSSV_CREATE]
THEN BEGIN

```

644 0839 4 IF .INSSGL_KFEADR EQL 0
645 0840 4 THEN
646 0841 5 BEGIN
647 0842 7 SIGNAL (((INSS_FAIL AND NOT STSSM SEVERITY)
648 0843 5 OR STSSK WARNING),
649 0844 5 2, LIST_DSC, INSSGQ_KFERN$, INSS_NOKFEFND);
650 0845 5
651 0846 4 RETURN TRUE;
652 0847 4 END;
653 0848 3
654 0849 4 ELSE
655 0850 4 BEGIN
656 0851 4 INSSGL_CTLMSK [INSSV_CREATE] = TRUE;      ' set create flag
657 0852 4 IF .INSSGL_KFEADR EQL 0
658 0853 4 THEN
659 0854 4 STATUS = INSSCREATE ();                  ! Create it
660 0855 4 ELSE
661 0856 4 STATUS = INSS_EXISTS;
662 0857 4
663 0858 4 IF NOT .STATUS
664 0859 5 THEN
665 0860 5 BEGIN
666 0861 5 SIGNAL (INSS_FAIL, 2, CREATE_DSC, INSSGQ_KFERN$, .STATUS);
667 0862 5
668 0863 5 IF .STATUS EQL INSS_INTRNLERR
669 0864 5 THEN
670 0865 4 SIGNAL (INSS_INTRNLERR, 1, .INSSL_INTRNLERR);
671 0866 3 END;
672 0867 3
673 0868 4
674 0869 3 IF (.INSSGL_CTLMSK [INSSV_LIST] OR .INSSGL_CTLMSK [INSSV_FULL])
675 0870 3 THEN REPORT (INSSLIST (.INSSGL_KFEADR));
676 0871 3 IF .INSSGL_CTLMSK [INSSV_GLOBAL]
677 0872 3 THEN REPORT (INSSGLOBAL (.INSSGL_KFEADR));
678 0873 < RETURN TRUE;
679 0874 1 END;          ! Routine dispatch

```

OFFC 00000 DISPATCH:

58	0000	CF	9E	00002	.WORD	Save R2, R3, R4, R5, R6, R7, R8, R9, R10, R11
5A	00000000G	00	9E	00007	MOVAB	DELETE DSC, R11
59	00000000G	00	9E	0000E	MOVAB	INSS\$LIST, R10
58	00000000G	00	9E	00015	MOVAB	INSS\$GL_REPLACE MSK, R9
57	00000000G	00	9E	0001C	MOVAB	INSS\$INTRNLERR, R8
56	00000000G	00	9E	00023	MOVAB	INSS\$GE_KFEADR, R7
55	00000000G	8F	D0	0002A	MOVL	INSS\$GQ_KFERNs, R6
54	00000000G	00	9E	00031	MOVAB	#INSS\$INTRNLERR, R5
53	00000000G	00	9E	00038	MOVAB	LIB\$SIGNAL, R4
07	00000000G	00	E8	0003F	MOVAB	INSS\$GL_CTLMSK, R3
		63	06	E0 00046	BLBS	INSS\$RFEFAB+8, 2\$
			01AA	31 0004A 1\$:	BBS	#6, INSS\$GL_CTLMSK, 3\$
4F	00000000G	63	06	E1 0004D 2\$:	BRW	26\$
			00	FB 00051 3\$:	BBC	#6, INSS\$GL_CTLMSK, 8\$
					CALLS	#0, INSS\$DELETE

			52	50	D0	00058	MOVL	R0, STATUS		0747	C			
			1E	52	E8	0005B	BLBS	STATUS, 4\$		0751	C			
				52	DD	0005E	PUSHL	STATUS		0750	C			
				56	DD	00060	PUSHL	R6						
				5B	DD	00062	PUSHL	R11						
			00000000*	02	DD	00064	PUSHL	#2						
				64	DD	00066	PUSHL	#<INSS FAIL&-8>						
				55	FB	0006C	CALLS	#5, LIB\$SIGNAL						
				52	D1	0006F	CMPL	STATUS, R5		0753				
				1A	12	00072	BNEQ	6\$						
				68	DD	00074	PUSHL	INSSL_INTRNLERR		0755				
				01	DD	00076	PUSHL	#1						
				55	DD	00078	PUSHL	R5		0758	C			
				0F	11	0007A	BRB	5\$			C			
			00000000G	8F	52	D1	0007C	4\$: CMPL	STATUS, #INSS_Removed		0757			
					09	12	00083	BNEQ	6\$					
					56	DD	00085	PUSHL	R6		0758	C		
					01	DD	00087	PUSHL	#1			C		
					52	DD	00089	PUSHL	STATUS			C		
					03	FB	0008B	5\$: CALLS	#3, LIB\$SIGNAL			C		
					01	E0	0008E	6\$: BBS	#1, INSSGL_CTLMSK+1, 7\$		0759			
					02	E1	00093	BBC	#2, INSSGL_CTLMSK+1, 1\$					
					7E	D4	00098	7\$: CLRL	-(SP)		0760			
					6A	FB	0009A	CALLS	#1 INSSLIST					
					01	4F	31	BRW	25\$					
					63	05	E0	000A0	8\$: BBS	#5 INSSGL_CTLMSK, 9\$		0766		
					03	0086	31	000A4	BRW	16\$				
					63	8F	88	000A7	9\$: BISB2	#64, INSSGL_CTLMSK		0778		
						69	D4	000AB	CLRL	INSSGL_REPLACE_MSK		0779		
			00000000G	00		00	FB	000AD	CALLS	#0, INSSDELETE		0780		
						52	DD	000B4	MOVL	R0, STATUS		0781		
						1F	52	E8	PUSHL	STATUS, 10\$		0786		
						52	DD	000B7	PUSHL	R6		0784		
						56	DD	000BA	PUSHL	STATUS				
						58	DD	000BC	PUSHL	R11				
						02	DD	000C0	PUSHL	#2				
						64	55	00000000*	PUSHL	#<INSS FAIL&-8>				
						05	FB	000C8	CALLS	#5, LIB\$SIGNAL				
						52	D1	000CB	CMPL	STATUS, R5		0788		
						09	12	000CE	BNEQ	10\$				
						68	DD	000D0	PUSHL	INSSL_INTRNLERR		0790		
						01	DD	000D2	PUSHL	#1				
						55	DD	000D4	PUSHL	R5				
						64	03	FB	000D6	CALLS	#3, LIB\$SIGNAL			C
						67	D4	000D9	10\$: CLRL	INSSGL_KFEADR		0792		
			00000000G	00		00	FB	000DB	CALLS	#0, INSSCLOSE FILE		0793		
			00000000G	00		00	FB	000E2	CALLS	#0, INSSOPEN FILE		0795		
						01	EF	000E9	EXTZV	#1, #1, INSSGL_CTLMSK+1, R0				
						50	F0	000EF	INSV	R0, #1, #1, INSSGL_REPLACE_MSK+1		0796		
						02	EF	000F5	EXTZV	#2, #1, INSSGL_CTLMSK+1, R0				
						50	F0	000FB	INSV	R0, #2, #1, INSSGL_REPLACE_MSK+1		0797		
						63	DD	00101	MOVL	INSSGL_REPLACE_MSK, INSSGL_CTLMSK		0798		
						63	10	88	BISB2	#16, INSSGL_CTLMSK		0799		
						63	8F	8A	BICB2	#64, INSSGL_CTLMSK		0800		
						40	67	D5	TSTL	INSSGL_KFEADR		0801		
			00000000G	00		00	FB	0010D	BNEQ	11\$				
						00	FB	0010F	CALLS	#0, INSSCREATE		0803		

			52	50	DO 00116	MOVL	R0	STATUS				
				07	11 00119	BRB	12\$					
			52 00000000G	8F	DO 0011B	11\$:	MOVL	#INSS EXISTS, STATUS	0805			
			20	52	E8 00122	12\$:	BLBS	STATUS, 13\$	0807			
				52	DD 00125		PUSHL	STATUS	0810			
				56	DD 00127		PUSHL	R6				
				F0	AB 9F 00129		PUSHAB	REPLACE_DSC				
				02	DD 0012C		PUSHL	#2				
			64 00000000G	8F	DD 0012E		PUSHL	#INSS FAIL		C		
			55	05	FB 00134		CALLS	#5, LIB\$SIGNAL	0812	C		
				52	D1 00137		CMPL	STATUS, R5	0814	C		
				09	12 0013A		BNEQ	13\$				
				68	DD 0013C		PUSHL	INSSL_INTRNLERR				
				01	DD 0013E		PUSHL	#1				
				55	DD 00140		PUSHL	R5				
			05 08	03	FB 00142		CALLS	#3, LIB\$SIGNAL	0817			
				01	E0 00145	13\$:	BBS	#1, INSSGL_CTLMSK+1, 14\$				
				02	E1 0014A		BBC	#2, INSSGL_CTLMSK+1, 15\$	0818			
				67	DD 0014F	14\$:	PUSHL	INSSGL_KFEADR				
				6A	01	FB 00151	CALLS	#1, INSSLIST				
				03	50	E8 00154	BLBS	STATUS, 15\$				
					0082	31 00157	BRW	23\$				
					0084	31 0015A	BRW	24\$				
			0A 05 1E 1A	50	67	DO 0015D	15\$:	MOVL	INSSGL_KFEADR, R0	0839		
				01	01	E0 00160	16\$:	BBS	#1, INSSGL_CTLMSK+1, 17\$	0834		
				02	A3	E0 00165		BBS	#2, INSSGL_CTLMSK+1, 17\$			
				04	A3	E1 0016A		BBC	#4, INSSGL_CTLMSK+1, 18\$	0835		
				04	63	E0 0016F	17\$:	BBS	#4, INSSGL_CTLMSK, 18\$	0836		
					55	12 00173	BNEQ	21\$		0839		
						00000000G	8F	DD 00175	#INSS_NOKFEFND	0842		
					56	DD 0017B	PUSHL	R6				
					BC	AB 9F 0017D	PUSHAB	LIST_DSC				
					02	DD 00180	PUSHL	#2				
					64	00000000*	8F	DD 00182	PUSHL	#<INSS_FAIL8-8>		
					05	FB 00188	CALLS	#5, LIB\$SIGNAL				
					6A	11 0018B	BRB	26\$		0845		
					63	10 88 0018D	18\$:	BISB2	#16, INSSGL_CTLMSK	0850		
					50	D5 00190	TSTL	R0		0851		
			00000000G	00	0C 12 00192	BNEQ	19\$					
				52	00	FB 00194	CALLS	#0, INSSCREATE		0853		
				52	50	DO 00198	MOVL	R0 STATUS				
				07	11	0019E	BRB	20\$				
			52 00000000G	8F	DO 001A0	19\$:	MOVL	#INSS EXISTS, STATUS	0855			
			20	52	E8 001A7	20\$:	BLBS	STATUS, 21\$	0857			
				52	DD 001AA		PUSHL	STATUS	0860			
				56	DD 001AC		PUSHL	R6				
				E0	AB 9F 001AE		PUSHAB	CREATE_DSC				
				02	DD 001B1		PUSHL	#2				
			64 00000000G	8F	DD 001B3		PUSHL	#INSS FAIL				
			55	05	FB 001B9		CALLS	#5, LIB\$SIGNAL				
				52	D1 001BC		CMPL	STATUS, R5	0862			
				09	12 001BF		BNEQ	21\$				
				68	DD 001C1		PUSHL	INSSL_INTRNLERR	0864			
				01	DD 001C3		PUSHL	#1				
				55	DD 001C5		PUSHL	R5				
			05 01	64	03	FB 001C7	21\$:	CALLS	#3, LIB\$SIGNAL			
				01	E0 001CA		BBS	#1, INSSGL_CTLMSK+1, 22\$	0868			

0D	01	A3	02	E1	001CF		BBC	#2, INSSGL CTLMSK+1, 24\$	
		6A	67	DD	001D4	22\$:	PUSHL	INSSGL KFEADR	0869
		05	01	FB	001D6		CALLS	#1, INSSLIST	
		64	50	E8	001D9		BLBS	STATUS, 24\$	
11	01	A3	01	FB	001DE	23\$:	PUSHL	STATUS	
		00000000G	04	E1	001E1	24\$:	CALLS	#1, LIB\$SIGNAL	
		00	67	DD	001E6		BBC	#4, INSSGL CTLMSK+1, 26\$	0870
		05	01	FB	001E8		PUSHL	INSSGL KFEADR	0871
		64	50	E8	001EF	25\$:	CALLS	#1, INSSGLOBAL	
		50	50	DD	001F2		BLBS	STATUS, 26\$	
		64	01	FB	001F4		PUSHL	STATUS	
		50	01	DD	001F7	26\$:	CALLS	#1, LIB\$SIGNAL	0873
					04	001FA	MOVL	#1, R0	
							RET		0874

: Routine Size: 507 bytes, Routine Base: \$CODE\$ + 0399

: 680 0875 1

```

: 682 0876 1 %SBTTL 'help';
: 683 0877 1
: 684 0878 1 ROUTINE HELP ( HLP_DSC ) =
: 685 0879 1 ++++
: 686 0880 1
: 687 0881 1 FUNCTIONAL DESCRIPTION:
: 688 0882 1 Conduct interactive help session
: 689 0883 1
: 690 0884 1 INPUT:
: 691 0885 1 hlp_dsc : Descriptor of command, possibly with qualifier(s) for
: 692 0886 1 which help is requested
: 693 0887 1
: 694 0888 1 OUTPUT:
: 695 0889 1 Interactive help session
: 696 0890 1
: 697 0891 1 ROUTINE VALUE:
: 698 0892 1 Low bit set for success
: 699 0893 1 unless an error is returned from lbr$output_help.
: 700 0894 1 ---
: 701 0895 2 BEGIN
: 702 0896 2 LOCAL
: 703 0897 2 HLP_FLAGS;
: 704 0898 2
: 705 0899 2 HLP_FLAGS = HLP$M_PROCESS OR HLP$M_GROUP OR HLP$M_SYSTEM;
: 706 P 0900 2 REPORT ( LBR$OUTPUT_HELP (LIB$PUT_OUTPUT, 0, .HLP_DSC,
: 707 0901 2 $DESCRIPTOR('INSTALHELP'), HLP_FLAGS, LIB$GET_INPUT ) );
: 708 0902 2 RETURN TRUE;
: 709 0903 1 END;

```

.PSECT \$PLITS,NOWRT,NOEXE,2

50 4C 48 4C 41 54 53 4E 49 00198	P.ACB:	.ASCII \INSTALHELP\	:
00000009	001A1	.BLKB 3	
00000000	001A4	P.ACA: .LONG 9	
00000000	001A8	.ADDRESS P.AC B	

.PSECT \$CODES,NOWRT,2

	0000 0000	HELP:	.WORD	Save nothing	0878
00000000G	0E DD 0002		PUSHL	#14	0899
	00 9F 0004		PUSHAB	LIB\$GET_INPUT	0901
04 AE 9F 000A			PUSHAB	HLP_FLAGS	
0000' CF 9F 0000D			PUSHAB	P.ACA	
04 AC DD 00011			PUSHL	HLP_DSC	
	7E D4 00014		CLRL	-(SP)	
00000000G 00	00 9F 00016		PUSHAB	LIB\$PUT_OUTPUT	
09	06 FB 0001C		CALLS	#6, LBR\$OUTPUT_HELP	
	50 E8 00023		BLBS	STATUS, 1\$	
00000000G 00	50 DD 00026		PUSHL	STATUS	
50	01 FB 00028		CALLS	#1, LIB\$SIGNAL	
	01 D0 0002F	1\$:	MOVL	#1, R0	0902
	04 00032		RET		0903

INSPRSOLD
V04-000

help

: Routine Size: 51 bytes, Routine Base: \$CODE\$ + 0594

: 710 0904 1
: 711 0905 1 END
: 712 0906 0 ELUDOM ! Module INSPRSOLD

L 2
16-Sep-1984 01:47:51
14-Sep-1984 12:35:40
VAX-11 Bliss-32 V4.0-742
[INSTAL.SRC]INSPRSOLD.B32;1

Page 27
(6)

XF1
VO4

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	16	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$SPLITS	428	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODE\$	1479	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	-----	Symbols	-----	Pages	Processing
	Total	Loaded	Percent	Mapped	Time
\$_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	22	0	1000	00:01.9

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:INSPRSOLD/OBJ=OBJ\$:INSPRSOLD MSRC\$:INSPRSOLD/UPDATE=(ENH\$:INSPRSOLD)

: Size: 1479 code + 444 data bytes

: Run Time: 00:25.3

: Elapsed Time: 01:40.8

: Lines/CPU Min: 2146

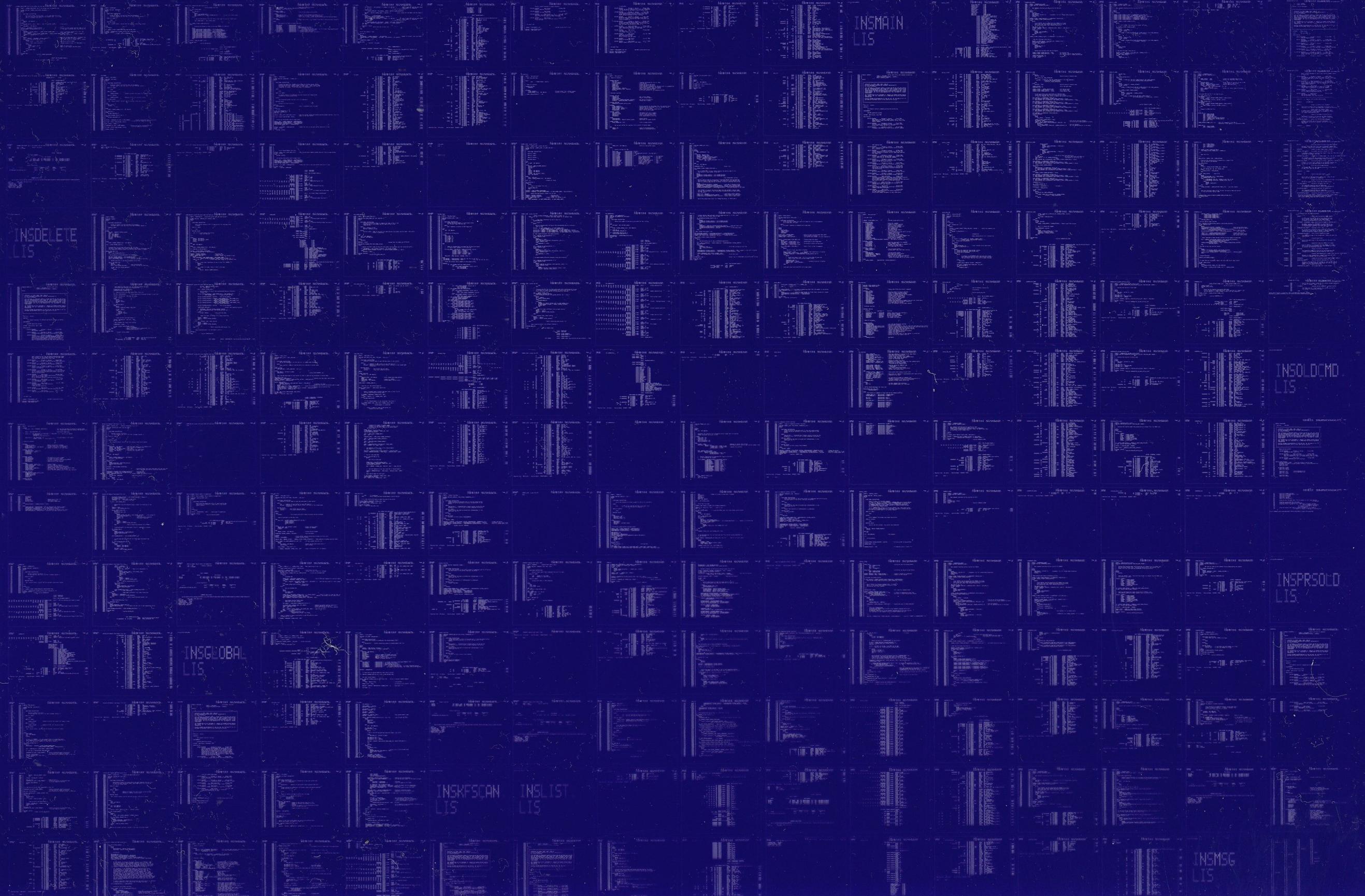
: Lexemes/CPU-Min: 15259

: Memory Used: 244 pages

: Compilation Complete

0189 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0190 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

XFDEF
FOR

DRSUP
LIS

IOSUP

ORDEF
MAR

LABUFFER
LIS

LASNOLDRO
LIS

JOBCTL
MAP

JOBCTLDEF
REQ

SYSQUEDEF
SQL

ACCOUNTING
LIS

LASWEER
LIS